**分支限界法解决0-1背包问题**

【代码】

#include <iostream>

using namespace std;

#define N 100

struct HeapNode {

int weight; //当前重量

int value; //当前价值

int level; //层级

int bound; //上界

int route[N];

};

//插入对堆元素，插入后还是最大值

void InsertHeap(HeapNode b[], HeapNode x, int & length)

{

int i;

for (i = length + 1; i > 1; i = i / 2)

{

if (x.bound <= b[i / 2].bound)

break;

else

b[i] = b[i / 2];

}

b[i] = x;

length = length + 1;

}

//删除堆顶元素，删除后还是最大值

void DeleteHeap(HeapNode b[], int &length)

{

int i;

b[1] = b[length]; //最后一个元素赋值给根

HeapNode temp = b[1];

b[length].bound = 0;

for (i = 2; i < length; i = i \* 2)

{

if (b[i].bound < b[i + 1].bound)

i++;

if (temp.bound >= b[i].bound)

break;

b[i / 2] = b[i];

}

i = i / 2;

b[i] = temp;

length = length - 1;

}

//返回堆中所有节点的最大上界

int MaxUpBound(HeapNode b[], int length)

{

int max = 0;

for (int i = 1; i <= length; i++)

{

if (max < b[i].bound)

max = b[i].bound;

}

return max;

}

void Pack01(int n, int w[], int v[],int W,HeapNode heap[],int & heapLength)

{

int i;

HeapNode rootNode;

rootNode.weight = 0;

rootNode.value = 0;

rootNode.bound = W \* (v[1] / w[1]);

for (i = 1; i < N; i++)

rootNode.route[i] = 0;

rootNode.level = 0;

InsertHeap(heap, rootNode, heapLength);

while (heapLength > 0)

{

HeapNode temp;

temp = heap[1];

int level = temp.level;

DeleteHeap(heap, heapLength);

if (level == n && temp.value >= MaxUpBound(heap, heapLength))

{

cout << "物品编号" << "\t" << "物品重量" << "\t" << "物品价值" << "\t" << "是否放入背包（1表示放入）" << endl;

for (i = 1; i <= n; i++)

cout << i << "\t\t " << w[i] <<"\t\t " << v[i] << "\t\t\t" << temp.route[i] << endl;//输出问题的解

cout << endl << "整个背包的总价值为:" << temp.value << endl;

return;

}

else

{

int j;

for (j = 0; j <= 1; j++)

{

if (temp.weight + j \* w[level + 1] <= W)

{

HeapNode node;

node.weight = temp.weight + j \* w[level + 1];

node.value = temp.value + j \* v[level + 1];

if (level <= n - 1)

node.bound = node.value + (W - node.weight) \*

(v[level + 1] / w[level + 1]);

else

node.bound = node.value;

for (i = 1; i <= level; i++)

node.route[i] = temp.route[i];

node.route[level + 1] = j;

node.level = level + 1;

InsertHeap(heap, node, heapLength);

}

}

}

}

}

int main()

{

HeapNode heap[N];

int heapLength = 0;

int n = 0,c = 0;

int weight[N], value[N];

FILE \*fp = NULL;

fp = fopen("D:\\Desktop\\test.txt", "a+");

if (fp == NULL) {

cout << "读取文件失败！" << endl;

}

fscanf\_s(fp, "%d", &n);

fscanf\_s(fp, "%d", &c);

for (int i = 1; i <= n; i++)

fscanf\_s(fp, "%d", &weight[i]);

for (int i = 1; i <= n; i++)

fscanf\_s(fp, "%d", &value[i]);

cout << "读取的物品个数：" << n << endl;

cout << "读取的背包的总容量：" << c << endl;

Pack01(n, weight, value, c, heap, heapLength);

system("pause");

return 0;

}

【效果】



